Ultra Low-Power Chip-to-Chip Communications

Design Review 1

ECE6332, Fall 2013

Ben Boudaoud and Chris Lukas

# Introduction and Motivation

In recent years general purpose processors (GPPs) and microcontroller units (MCUs) have been targeted for reduced power consumption as part of an effort to increase efficiency and correspondingly extend battery life in consumer electronics. This reduced power consumption has been enabled by the use of multiple power and clock domains, the introduction of low power reduced functionality modes, sub-threshold operation, along with other techniques. In addition, reduced design and manufacturing cost of application specific integrated circuits (ASICs) results in the ability to search for new research areas in power and efficiency further pushing the performance of these ultra-low power systems.

With power consumption of complete systems on chip (SoCs) entering the microwatt to nanowatt range, circuit components which have traditionally not been dominant consumers in the system power budget have become increasingly relevant due to progress in other areas in the SoC. One of these components is low-throughput, serial, inter-chip communication, where little innovation has occurred over the past decade. Custom solutions are strong contenders in this space as they can be tailored to given device specifications in order to minimize power consumption. However, these custom interfaces lag severely in one critical metric, widespread adoption. As a result it is critical that the design trade-off between customization and interoperability be carefully considered when designing new standards for inter-chip communications. Our plan is to design a custom, low power, energy efficient chip-to-chip communication scheme allowing for ultra-low power SoCs to communicate off-chip without said communication significantly impacting their power budgets.

# Background Research

Though a number of papers related to high-speed inter-chip communications have been published, relatively little information related to ultra-low power/throughput is available in traditional research journals. Thus a number of papers motivating the need for lower power chip-to-chip interfaces, discussing various physical layer specifications for general purpose serial I/O, and describing common serial transfer structures and topologies were surveyed. Though this survey does not provide the “state of the art” per say, it does give a set of useful guidelines for considering the problem of ultra-low power serialized I/O design.

# Progress

Currently we have investigated a great deal of the physical layer of the chip-to-chip communications design problem. We have researched and modeled a typical communication channel and done some minimal investigation of power, area, and frequency for operation of buffers on this channel. In addition to buffer and channel concerns, latch and register design for the serializing shift register has been approached and iterated several times. These three components together (buffers, channel, and shift register) make up the majority of the physical layer consideration for latching data in and out of the module.

By the time the project proposal is due we hope to have a majority of the physical specification completed for our protocol. Though we are not yet sure whether we will be creating a bit compliant protocol or offering a higher power “compatibility mode”, we will most likely decide to be in some way compliant with the serial peripheral interface (SPI) standard. In addition, we hope to decide on a technology (IBM 0.13um or FreePDK 45nm) and produce some semi-accurate standby and active power measurements of our actual register and driver cells by the time the proposal is due. We expect the largest consumers of power to be the number of wires used in the chip-to-chip communication, the fact that standard SPI uses a full swing on the transmission, and the activity factor of the transmission lines. Luckily these are knobs that we can change in our design in order to reduce the total power consumption. However, the more these change, the less similar the design is to the SPI standard, and we will be considering this trade-off.

After the proposal we will continue the design of our interface with the creation of arbitration blocks and power management as we see fit. Ideally this additional work will be done in Verilog and synthesized using a standard cell library rather than hand schematic entry. If this is the case, the decision to synthesize this logic (as opposed to designing it by hand) will be justified with power trade-offs and cost-benefit analysis. Some common interface features (such as split TX and RX buffers, auto-transmission on TX register write, and control register for state retention and peripheral bus management) may be added, time permitting, and tested in a behavioral simulator such as ModelSim. Last but not least, SPICE simulation of this full circuit will be used to extract accurate energy and power consumption for comparison against the state of the art, both in energy per bit, and average power when running with a realistic use factor.

# Paper Summaries

*A Batteryless 19 uW MICS/ISM-Band Energy Harvesting Body Sensor Node SoC for ExG Applications – Zhang et al.*

The UVA BSN SoC is a low power chip for monitoring ECG, EEG, and EMG data while being self-powered. One of the major challenges with chips like this is nonvolatile storage and inter-chip communication. The paper published in January of 2013 does not cover a solution to either of these issues, and serves as a possible candidate for our planned design. Nonvolatile storage consumes relatively high amounts of power during the read and write stages, meaning that a self-powered SoC can die while writing vital data. Inter-chip communication is also usually a high power function when using standard protocols, however not much research has been done in low power inter-chip communication.

This paper will help us in understanding power constraints, as it is a real SoC and the paper contains information on relative power consumption between different components in the SoC. The paper also gives us an idea of what type of data is going to be sent across our design, meaning that we won’t be sending large amounts of data at one time, ruling out the possibility for a high speed / low energy per bit design, as these designs generally contain a large overhead resulting in this high efficiency only with large amounts of data. This paper also helps us understand what type of speed of communication we need to design around, as we will probably not be designing with a higher speed than the clock speed of this device, but may go lower if our simulations lead us to doing so.

*Phoenix: An Ultra-low Power Processor for Cubic Millimeter Sensor Systems –Seok et al.*

The Phoenix system-on-chip (SoC) consists of an ultra-low power core and SRAM banks, along with a system bus containing a timer and temperature sensor implemented in a 0.18um CMOS process. The platform designers target power reduction via a holistic design methodology in which they consider aggressive standby duty cycling (>99% of lifetime in inactive mode) to be a normal part of operation. The suggested peripherals (timer and temperature sensor) are also chosen and specified for this ultra-low active time, resulting in the SoC being optimized for bursty wake-and-measure deployments. As a result of these ultra-low active-to-standby ratios and standby powers, average system power consumption in the range of 50pW is made possible by a combination of duty cycling, supply reduction, and leakage minimization in common SoC circuits. This is relevant as it frames the overhead power of operating a chip-to-chip interface as one of the most significant power consumers in a system (even if at a relatively low duty active ratio).

While the authors do make use of near or sub-VT operation for logic, they primarily focus on reducing leakage in standby through effective power gating methodologies and SRAM bit cell leakage reduction. Both of these optimizations are seen as critical as these leakage currents are a static cost for the system (unlike core logic, which is power gated during inactive periods). This paper does an excellent job of considering the entire problem of designing an SoC for efficient low power applications, which often use extremely low active ratios (<0.1%) to reduce average power consumption. This is a good example of how often constraints that occur at the application level may have impacts that reach all the way down to the hardware designer.

*Low-Voltage Low-Power LVDS Drivers –Chen et al.*

This paper focuses on the driver design for low voltage differential signaling (LVDS) communications. The basic principles of LVDS, including differential signaling, bus termination, and common voltage levels are all covered. The paper cites the complexity and cost of IC packaging and PCB integration for large parallel buses as the motivating factor for throughput to pin ratios in the Gbps/pin and above.  The authors present a typical CMOS implementation of an LVDS driver circuit and discuss the issue of switching headroom for PMOS devices at supply voltages of less than 2.5V.

Two approaches to solving this headroom problem are proposed. Double current source (DCS) drivers use two PMOS current sources in the pull-up network and require the NMOS transistors to sink twice the required current when conducting. Switchable current source (SCS) drivers use two switchable PMOS current sources in place of PMOS FETs allowing for operation near or at the minimum required current for LVDS. The DCS and SCS drivers were both fabricated in TSMC 0.35um CMOS process and are evaluated using an eye diagram over a pseudo random binary sequence (PRBS) at 680Mbps and 1Gbps. Though the DCS driver has nearly twice the static current consumption (23mW) it significantly outperforms the SCS driver at both operating frequencies.

*Aspects and Solutions to Designing Standard LVCMOS I/O Buffers in 90nm Process – Kannan et al.*

This paper summarizes design principles and challenges for split core and I/O voltage design. The author discusses the source and impact of simultaneous switching noise (SSN) in circuit operation and details the effect on buffer design. The practice of using thicker gate-oxide FETs to supporting legacy voltage levels while continuing to scale core voltage levels is introduced, and a simple level conversion topology for interfacing the two is introduced. In addition, a control logic and pre-driver circuit with split P and N drive outputs to the final driver stage is proposed, allowing for tri-state control of the output pin.

A crude (order of magnitude) lumped element model for the parasitics of a pad, bond wire, and pin is proposed based upon industry standards. The proposed driver structures are created in a 65nm CMOS process and evaluated with 10pF load cap at 200MHz operating frequency. A brief discussion of results is included. This paper provides a useful outline of design principles for general LVCMOS I/O interfaces, but did not lend a significant amount of insight into any novel practices for buffer design or optimization.

*Low Power Methodology Manual: For System-on-Chip Design*

This book talks about low power methodologies for designing circuits. The book summarizes power gating for registers and how to retain data when power gating. This is done by keeping power to parts of the registers responsible for holding data while removing the power to the rest of the circuit. This is important for a design like SPI where you won’t always be sending or receiving data. I plan to use this source mainly for low power register design.

*Serial Bus Encoding for Low Power Application – Saneei et al.*

This paper discusses the use of serialization and data encoding for reduction of parasitic losses due to inter-wire capacitance in on-chip buses. The authors observe that by partially or completely serializing operation on these buses, the designer can significantly reduce the power that may be lost to large inter-wire capacitance in a fully parallelized case. As a further step towards reducing inter-wire cap (and therefore switching power) on parallel buses, a differenced (XOR-based) encoding scheme is proposed to exploit the temporal coherence of data and greatly reduce bus activity for low input stream entropy.

  Though a good deal of this paper was dedicated to the minimization of on-chip bus energy, not inter-chip communications energy, it held some valuable lessons for designing low power off-chip communications as well. The possibility of offering a simple encoding mode to greatly reduce output pin activity factor could mean serious power reduction for applications in which the input bit stream may be large, but of relatively low entropy.

*FPGA Implementation of I2C and SPI Protocols: A Comparative Study – Oudjida et al.*

This paper compares the features of SPI and I2C, weighing in their pros and cons. It explains that I2C has a higher complexity, leading to a higher amount of area in layout. The paper also specifies that I2C is limited to 3 specific frequencies, while using SPI allows you to choose a frequency that works best for your design. This would allow us to find an optimal energy per bit, further reducing energy consumption in SPI. In addition the paper describes I2C as having two pull up resistors, which increase the static power consumption, further leading us to base our design on SPI.

This paper lead us in the direction of developing our design to be more similar to SPI than I2C. This is because of the comparison of power consumption between the two standard designs. In addition, I2C is designed for support of multiple master devices, which is not a desired application in our design. We’re not yet certain whether we will be SPI compatible, but we are leaning towards a custom design similar to SPI, as we hope to see a dramatic reduction in power costs over SPI by doing so.

# Simulation

Several key structures were modeled and simulated in Cadence for the purpose of proof of concept and back-of-the-envelope parameter estimation. A number of these simulations and their results are summarized below.

*Chip-to-chip Channel Model*

The communication channel between two integrated circuits (ICs) can be a complicated one to model. Drivers, ESD protection, bond wires, pins, and a micro strip PCB transmission line are all included in this schematic.

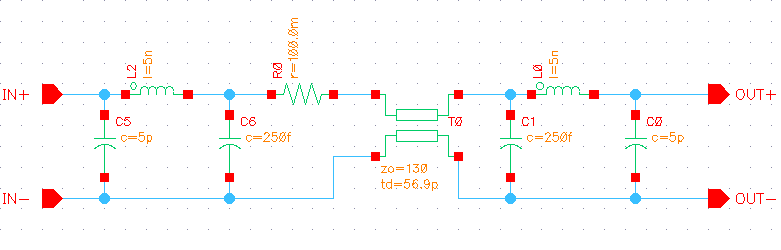


Figure 1: Chip-to-chip communications channel

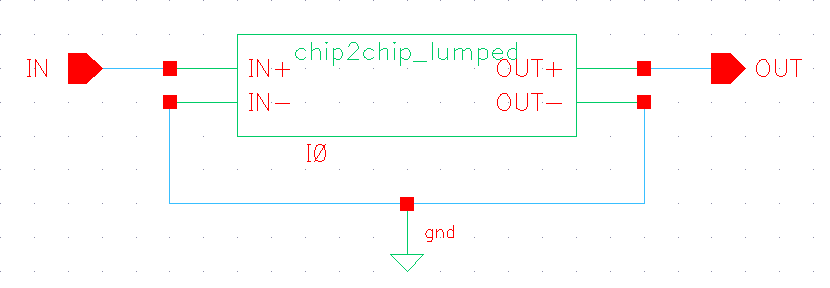
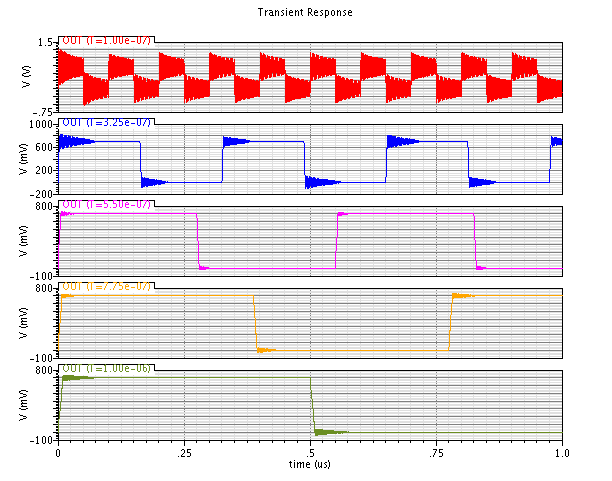


Figure 2: Lumped chip-to-chip communications test bench

The channel model was evaluated under 3 metrics: operating frequency, physical length, and voltage. The results are included below.

*Channel Performance at Various Operating Frequency*

The channel model proposed in Figure 1 will behave as a low-pass filter over much of the operating region. Thus, as operating (switching) frequency increases we expect the voltage at the output of the channel to be of increasingly poor fidelity, until recovering the initially transmitted data is no longer an option. This is verified in the simulation performed below (physical length set to 5cm and voltage to 1V peak-to-peak. Though 10MHz may not be the actual cutoff for operation, it can be seen that at this point the channel is nearing VM for a typical inverter with edge bounce, which will cause glitching at the receive buffer’s output.



3.07 MHz

1.8 MHz

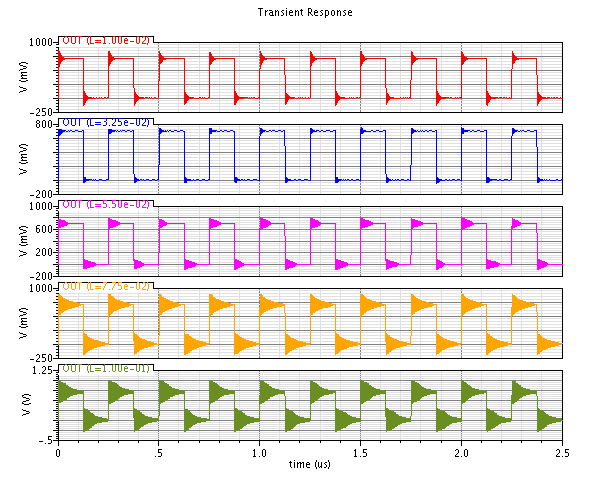
1.3 MHz

1 MHz

Figure 3: Channel Performance at Various Operating Frequencies

*Channel Performance at Various Physical Length*

In addition to the operating frequency of the channel, the physical length of the channel (largely determined by the PCB trace length) will have a significant effect on communications. Thus we sweep the length of the transmission line from 1 to 10cm, with operating frequency of 4MHz and voltage swing of 1V. Since as the trace grows longer it contributes greater parasitics to the lumped element model, we see increased edge bounce effects similar to those of the previous simulation.



3.25 cm

1 cm

5.5 cm

7.75 cm

10 cm

Figure 4: Channel Performance at Various Physical Lengths

*Channel Performance at Various Operating Voltage*

The voltage swing of communications will have an effect on maximum throughput and power consumption of the input/output buffer. Since the chip-to-chip channel model is fairly lossless (contains fairly little series or shunt resistance) we do not expect voltage level to have a significant effect on communications. It is worth noting that these voltage levels are produced as ideal stimuli and therefore have infinite drive strength. Thus, this simulation does not account for the reduction of throughput due to weak drive strength over any part of a bit transition.

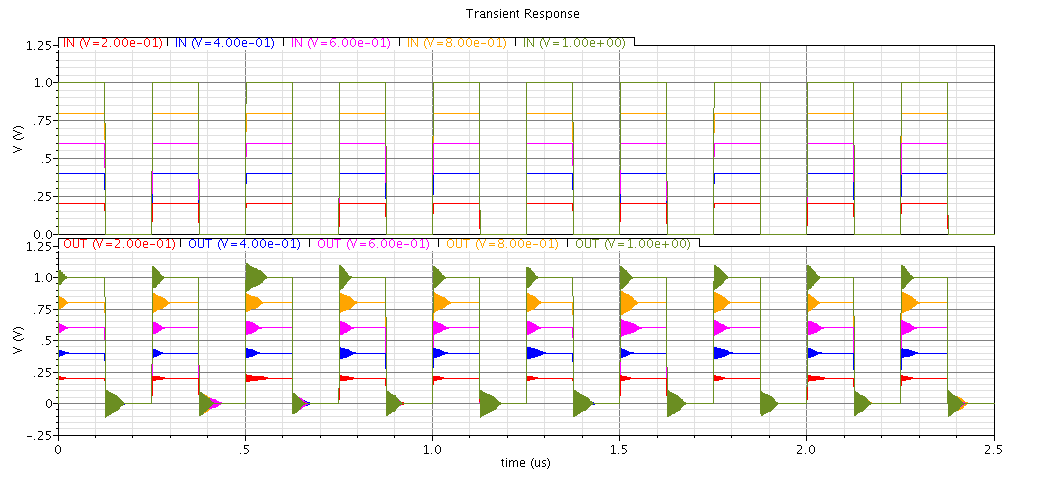


Figure 5: Channel Performance at Various Operating Voltage

### Buffer Size and Constraints

Though a final buffer design has not yet been arrived at for this project. A first-pass parametrized, two-inverter buffer was created for some simple testing of order of magnitude for power consumption and sizing constraints. The buffer design and test bench schematics are provided below. It may be worth noting that in the buffer test bench the output swing (specified by VDD-VSS of the buffer) is fixed at 1V.

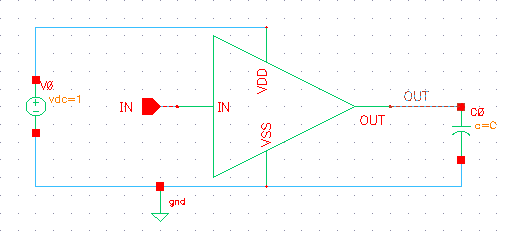


Figure 6: Buffer Test Bench

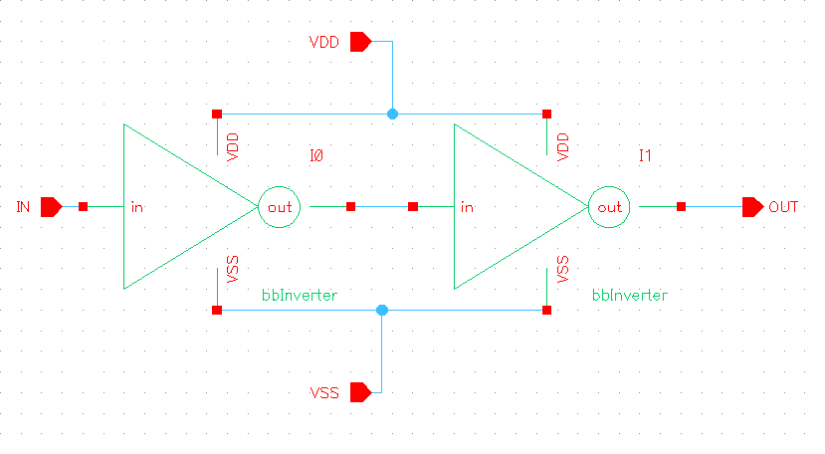
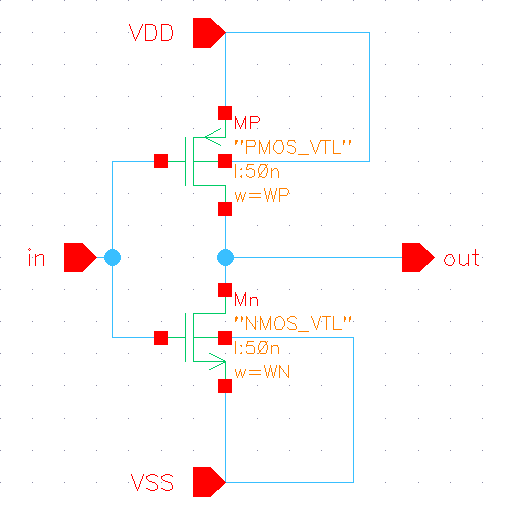
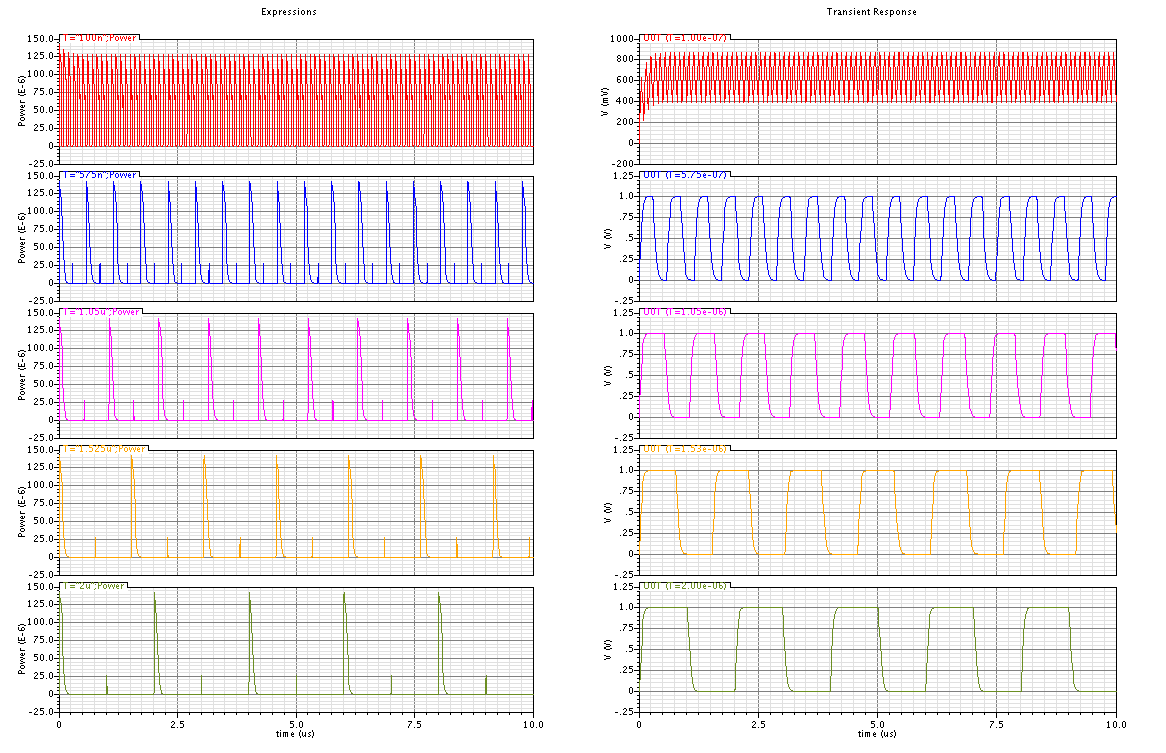


Figure 7: Simple 2-Inverter Buffer Design and Parametrized Inverter

*Buffer Frequency Sweep*

The buffer input frequency was swept from 500 kHz to 10 MHz, with 10pF of load cap and the resulting output waveform and power consumed from VDD were plotted in the following simulation results.



10 MHz10 MHz

3.07 MHz

1.8 MHz

1.3 MHz

1 MHz

Figure 8: Buffer Power and Output at Various Operating Frequency

*Buffer Output Capacitance Sweep*

As both a sanity check and an evaluation of this dual-inverter design the buffer circuit was evaluated at 1MHz for a sweep of load cap in the range of 1-50pF. The resulting time-domain power and output voltage waveforms are provided in the figure below.

1 pF

13.2 pF

25.5 pF

37.8 pF

50 pF

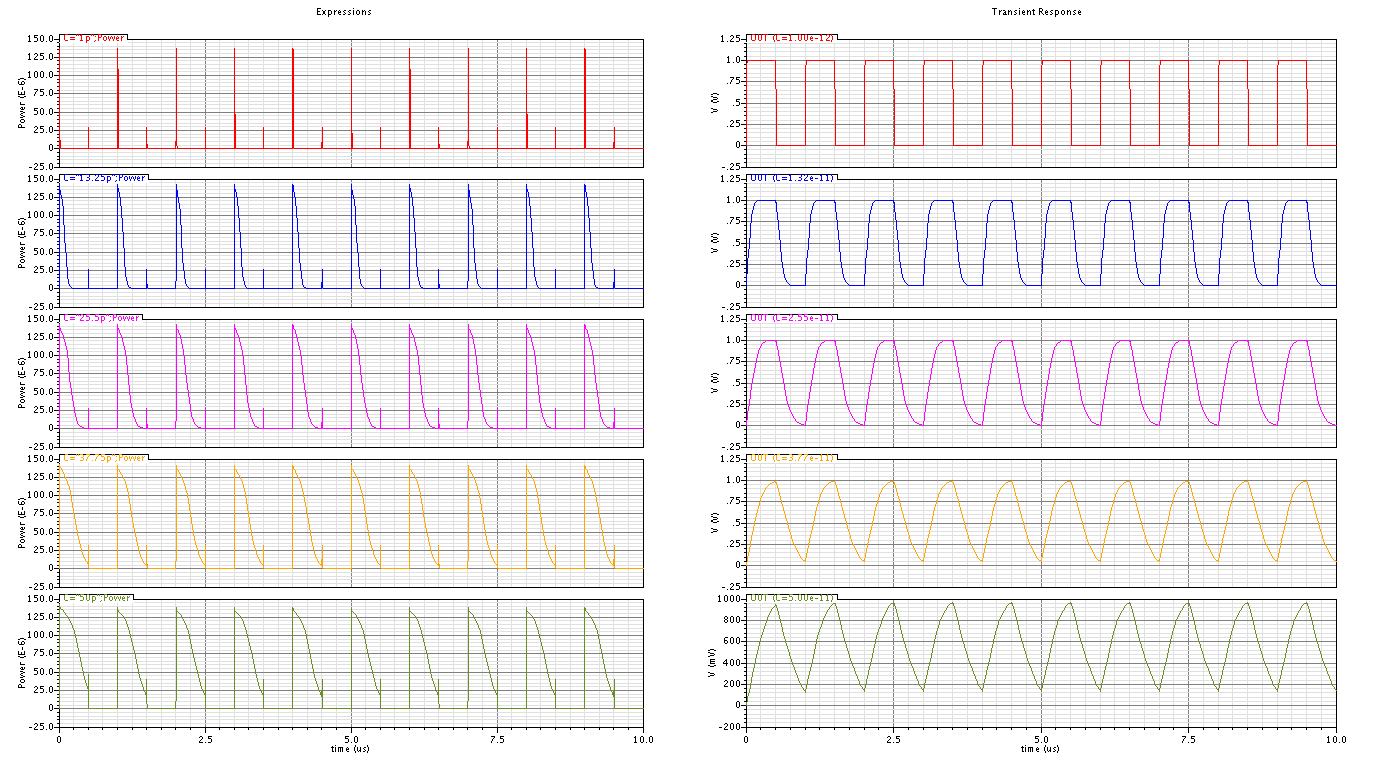


Figure 9: Buffer Power and Output at Various Load Cap

*Buffer Sizing Sweep*

As part of a crude effort to find appropriate range for the sizing of the final output driver in our transmitter/receiver circuit the size (width of the NMOS and PMOS devices) in the buffer were swept from 90nm to 720nm. Operating frequency was fixed at 1MHz and the load cap to 10pF for the sake of this simulation. Plotted results are included below.

Figure : Buffer Power and Output at Various Buffer Size

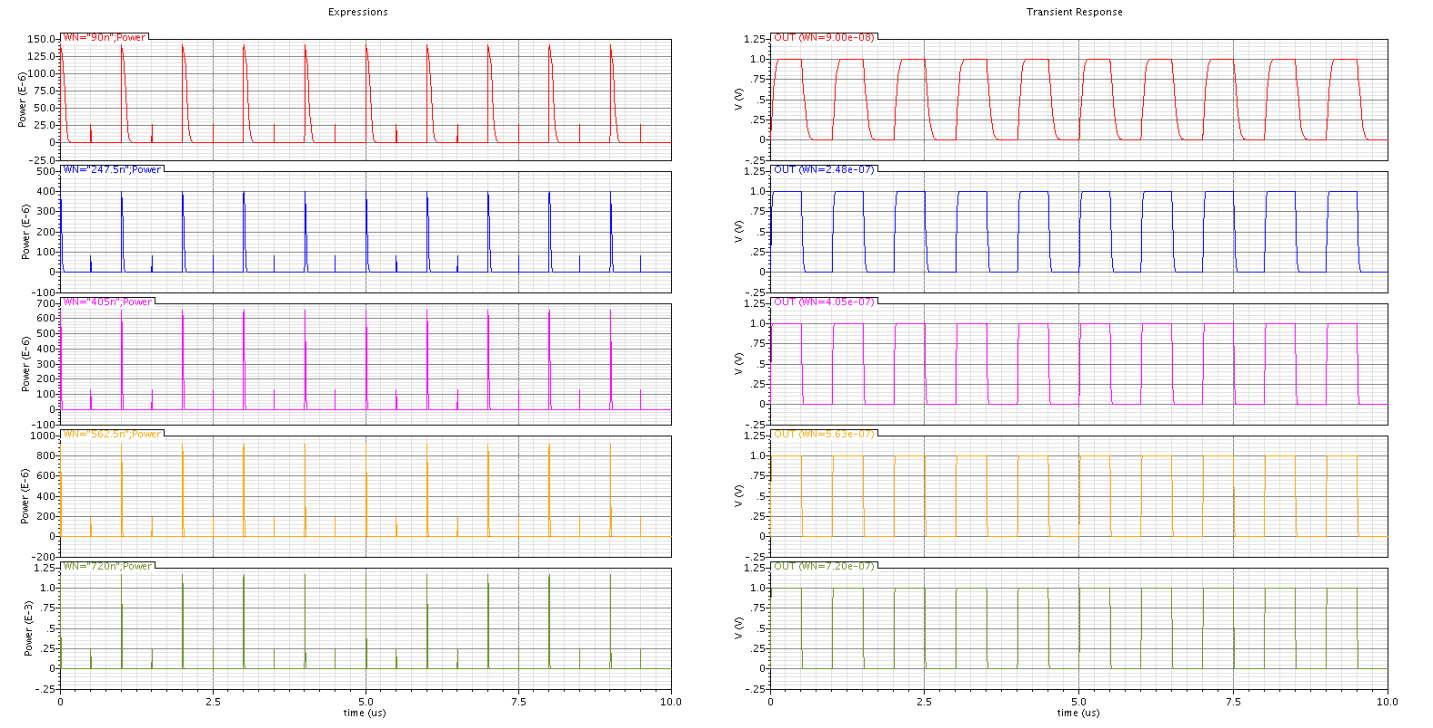
90 nm

248 nm

405 nm

563 nm

720 nm



### Two Buffers and Channel

To provide a somewhat representative assessment of the transceiver circuit operating in context the following test bench was created using the communication channel model and the buffer designed in the previous simulation.

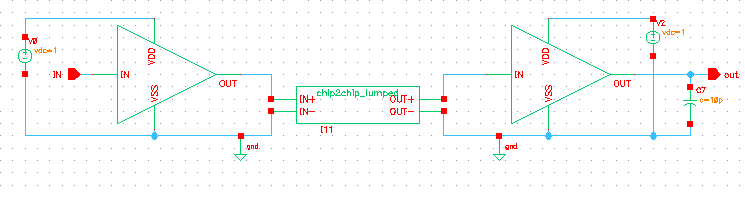


Figure 11: Channel and Buffer Test Bench

*Buffer Size versus Output Voltage and Power Consumption*

Much like in the previous buffer-only simulation a sweep of the parameterized buffer width was performed and output voltage and input power were measured. However, in this case the buffer’s output load was made up of the communication channel, another identical buffer’s input capacitance, and an output load cap of 10pF. The power recorded was that of the single buffer (two inverters) operating at 1 MHz.

90 nm

317 nm

545 nm

773 nm

1 um

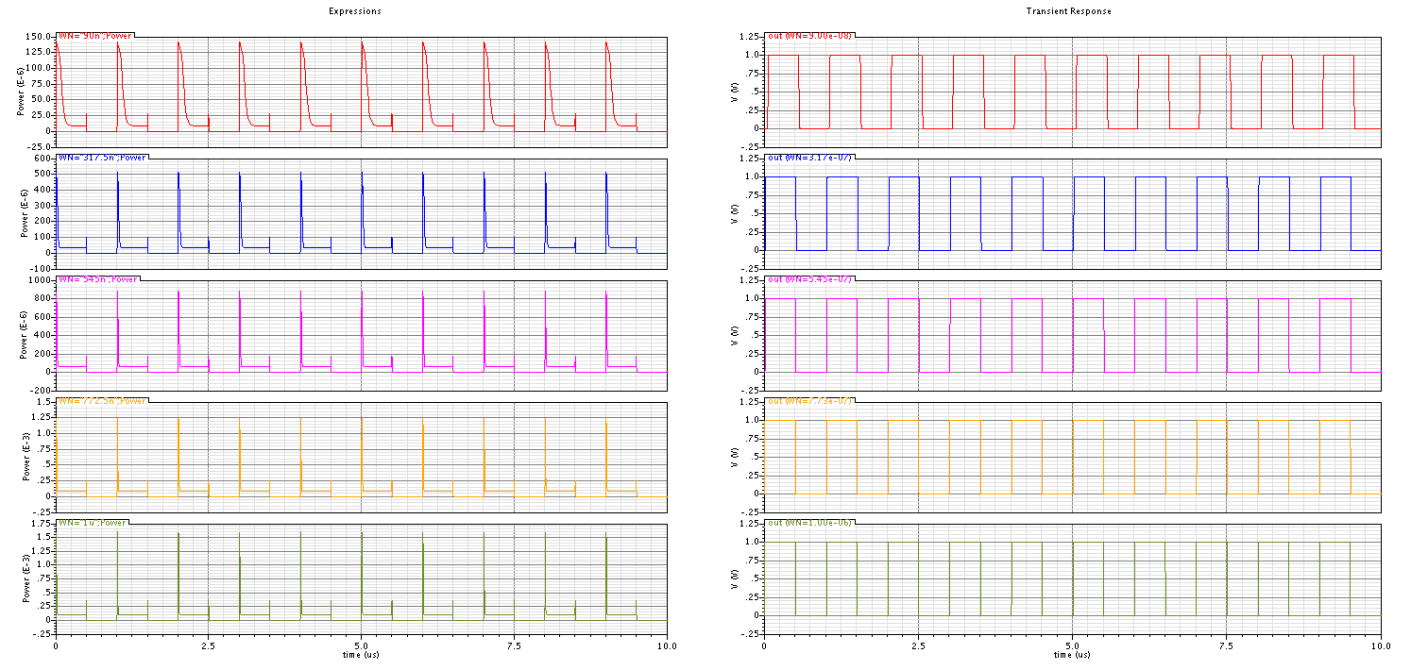


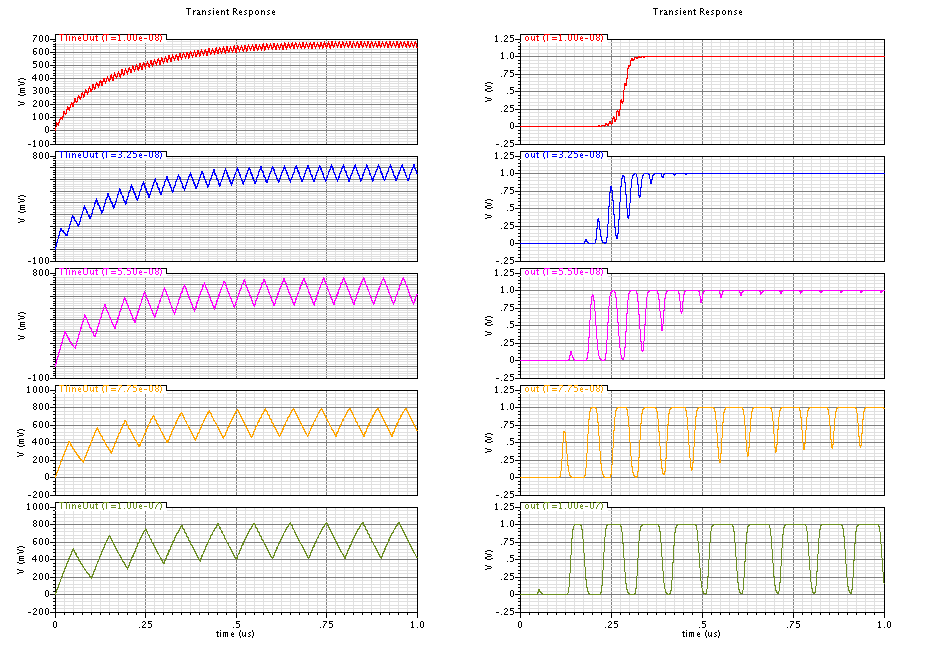
Figure 12: Output Buffer Voltage and Input Buffer Power at Various Buffer Size

*\*

*Transmission Line and Buffer Output at Various Frequency and Buffer Size*

As part of a slightly more complex simulation the operating frequency of the buffers/channel was swept from 10 to 100 MHz for 3 different widths of the NMOS transistor in the two inverter buffer (90nm, 180nm, and 360nm). The PMOS width is determined statically as twice the NMOS width for this sweep. The output load cap was fixed to 10pF and the following results were produced.

*90nm NMOS (180nm PMOS)*



100 MHz

30.8 MHz

18.2 MHz

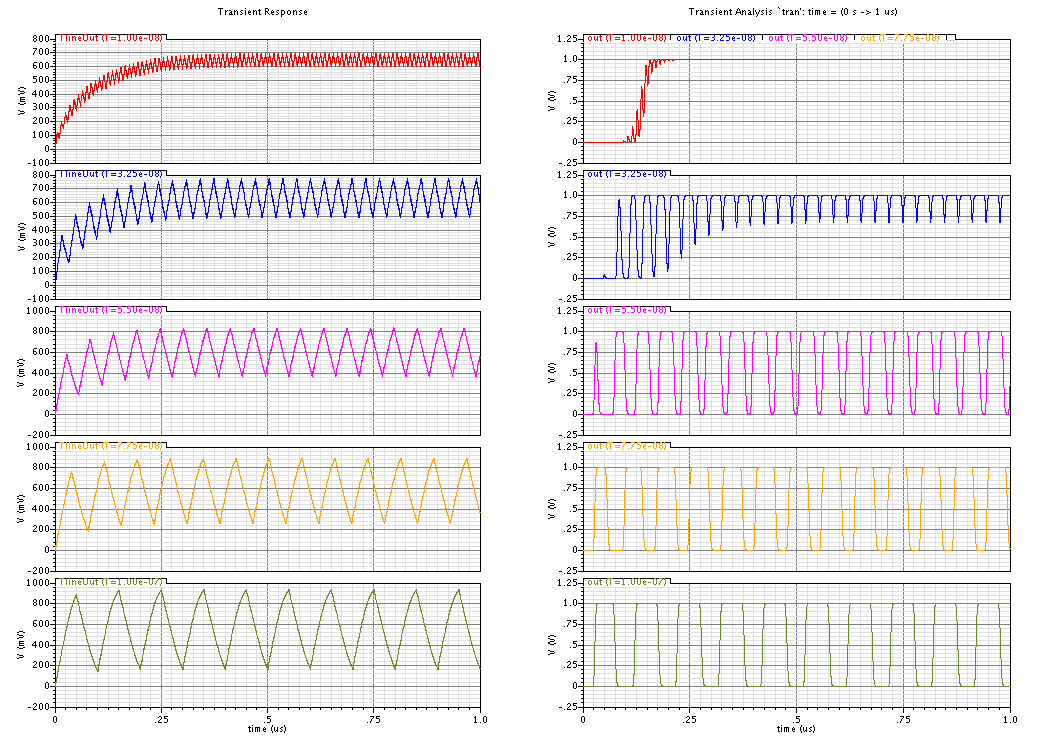
12.9 MHz

10 MHz

Figure 13: Output Buffer Voltage and Current at Various Operating Frequency (NMOS width of 90nm)

For the 90nm (minimum-sized) buffer we see that performance is significantly effected all the way down to near 10 MHz. Though we may be able to operate sufficiently in this ultra-low throughput regime with a minimum sized buffer it is probably worth noting that a practical design should be well out-performing this one, well above its intended frequency of operation.

*180nm NMOS (360nm PMOS)*



100 MHz

30.8 MHz

18.2 MHz

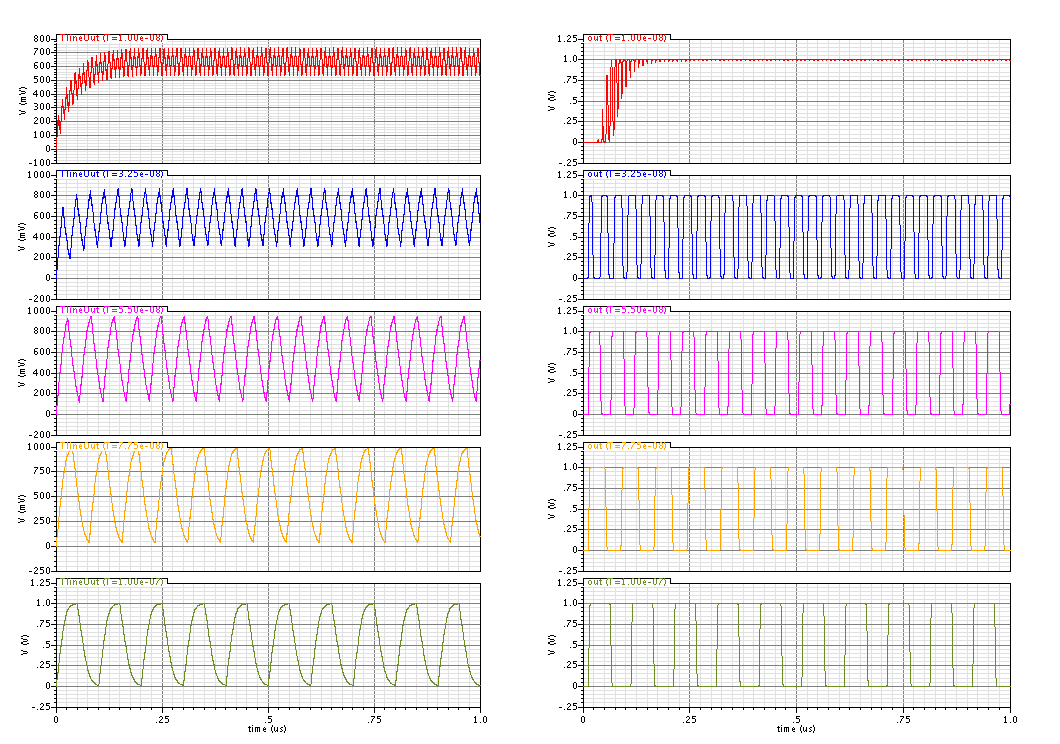
12.9 MHz

10 MHz

Figure 14: Output Buffer Voltage and Current at Various Operating Frequency (NMOS width of 180nm)

The 180nm (double minimum-sized) design performs slightly better, but still has significant output drive and input buffering issues at 30 MHz. Since this frequency will most likely be considered outside of the desired range of operation, this circuit may work fine, but is somewhat close to the minimum absolute tolerances for a transceiver working at 10MHz.

*360nm NMOS (720nm PMOS)*



100 MHz

30.8 MHz

18.2 MHz

12.9 MHz

10 MHz

Figure 15: Output Buffer Voltage and Current at Various Operating Frequency (NMOS width of 360nm)

The 360nm (quadruple minimum-sized) design seems to perform more than adequately. Its output is still full swing at 30 MHz and seems to perform acceptable both in drive strength (pushes the channel above/below VM at the driver) and output buffering (full swing at the output of the channel buffer). Thus, for future buffer design exploration NMOS devices of width 300nm and above (PMOS of width 600nm and above) will be considered.

### Shift Register Design

The shift register is the main component feeding data from one chip to another and will require much investigation into an optimal low power design. The design I started with contains an 8 bit register containing D flip flops. These flip flops were designed using clocked inverter latches; however we are still studying other latch designs and their tradeoffs.

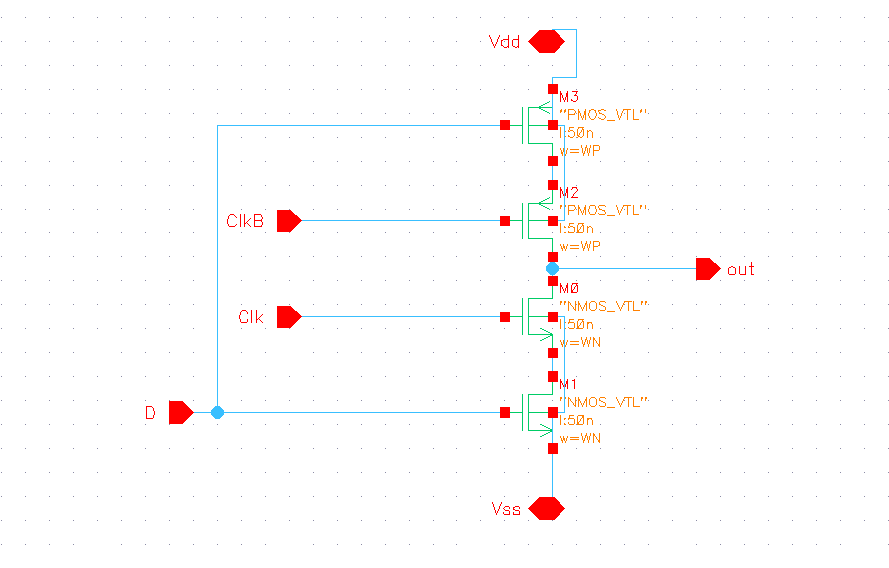
  
Figure : An inverter controlled by the clock

Figure 16 shows our current clocked inverter design. Notice the clock transistors on the inside of the D signal transistors. This disallows AC anomalies to travel from the D signal to the output while the clock is low.

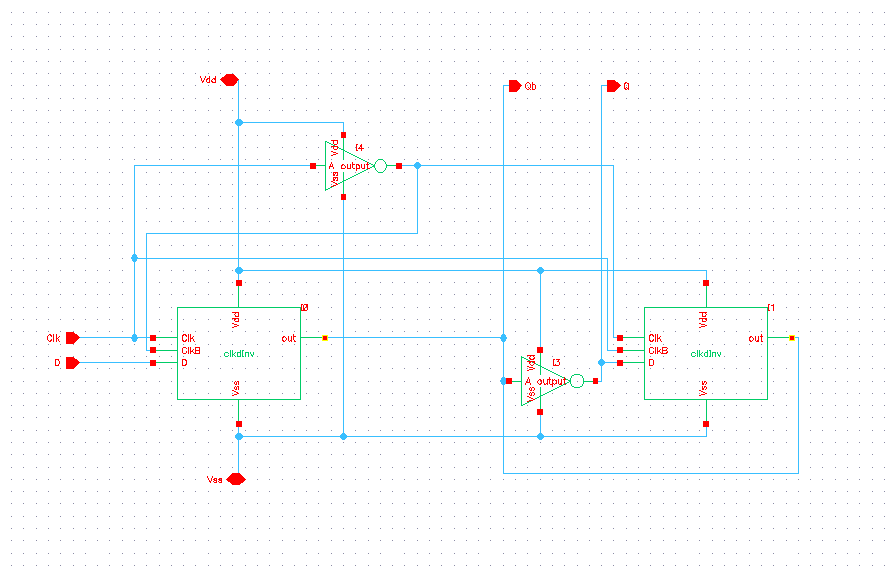
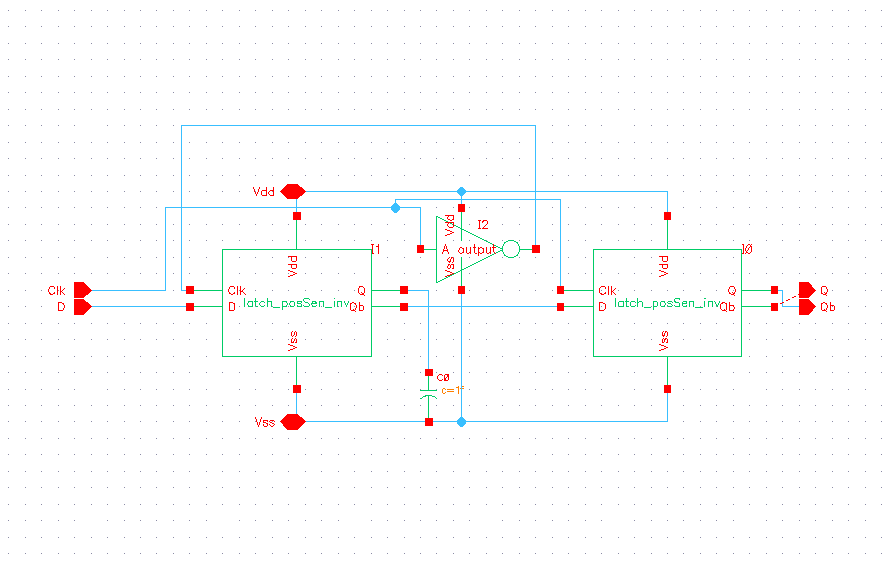
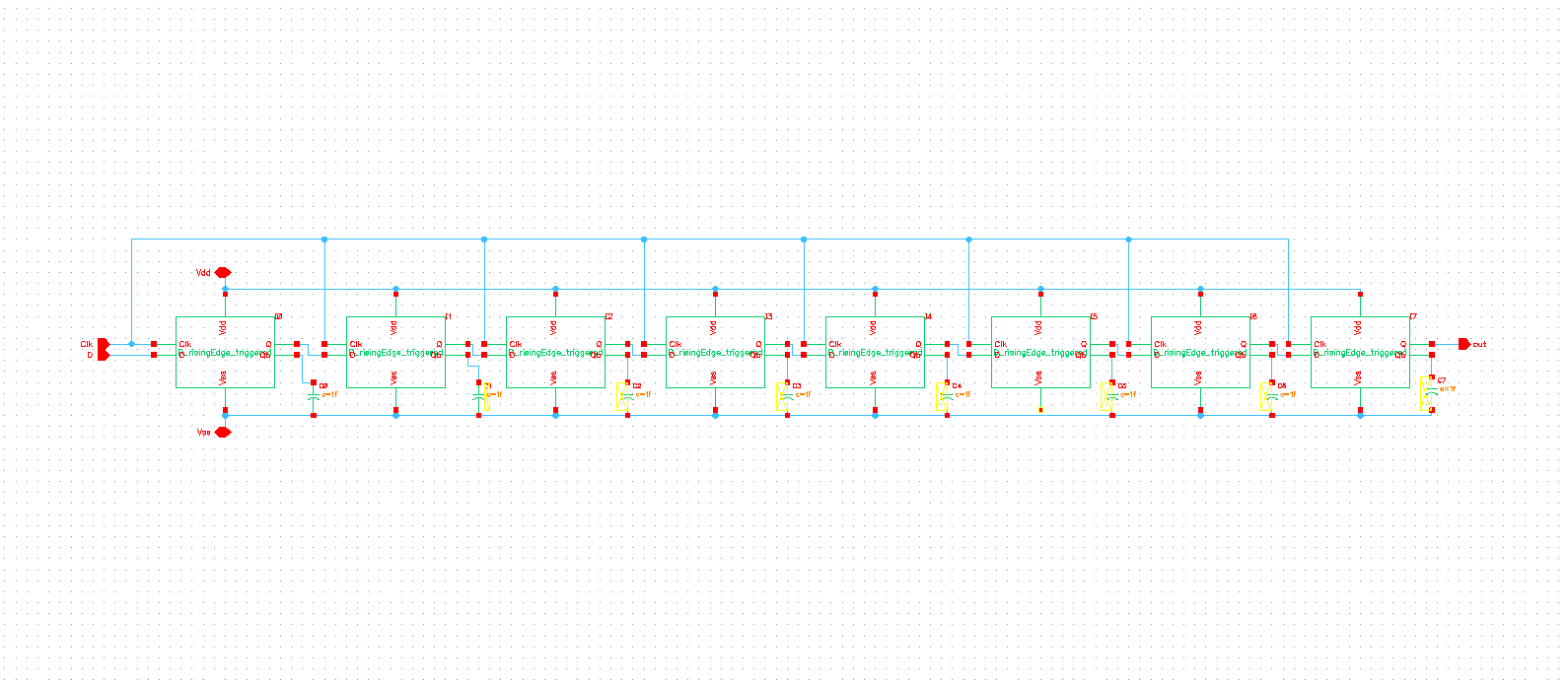


Figure : An positive sensitive inverting latch created using two clocked inverters and two inverters

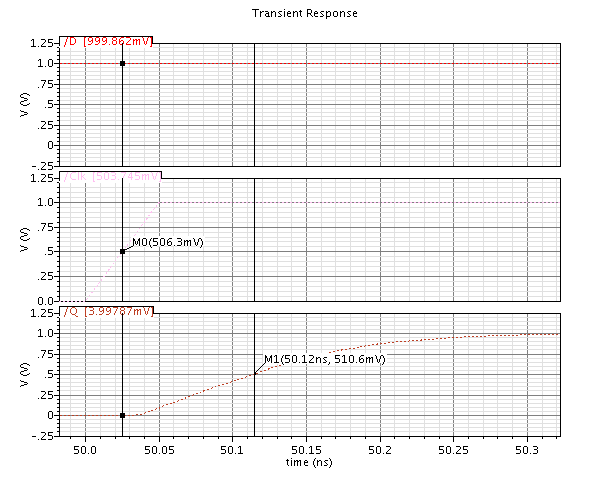
This is the design of our inverting latch. This design will not change regardless of our latch design, allowing us to optimize our latch while keeping functionality at the higher levels of abstraction. Even though our latch inverts, the higher level design stays the same because a flip flop requires two latches, so the output stays the same with inverting latches or non-inverting latches, because the amount is even.

Figure : A positive edge triggered D flip flop

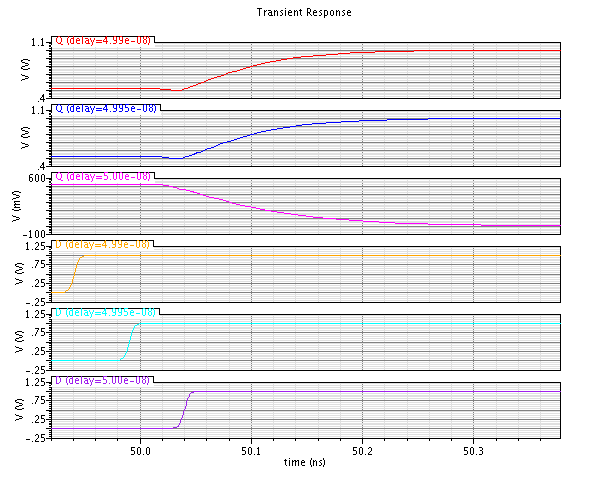
Our flip flop design is straight forward, containing two latches in series, with the first containing an inverted clock while the second contains the non-inverted clock, making it a positive edge triggered flip flop. If both latches contained a same phase clock the data coming in D could travel through the entire block, resulting in a non-functional design.

Figure : An 8 bit shift register containing our D flip flop design

The purpose of a shift register is to be able to shift in a certain amount of bits at once in parallel, and then shift them all out through the end in multiple clock cycles. This is how the chip communicates with another chip. If an even amount of D flip flops are used in the shift register, they can be connected by Qb instead of Q, inverting the bit with each clock cycle. This will reduce the propagation delay from one flip flop to the next (the Clk to Q), but may not help us in our design, as it may not create a reduction in power.

Figure : The simulation used to calculate Clock to Q delay in our flip flop design

A simple simulation resulting in an important metric in flip flop design is the Clock to Q simulation. It only requires setting the input at a Boolean 1 value while waiting for the clock to rise. You then measure the 50% point in the clock transition to the 50% point in the Q value and find the difference, resulting in the Clock to Q value, which we calculated to be 95 ps.

Figure : A parametric simulation of setup time

This figure shows a parametric simulation of multiple D values and Q values that either latch to the correct value, or do not latch in time. Finding the final simulation to latch results in the setup time. We found the setup time of our D flip flop to be 42 ps.

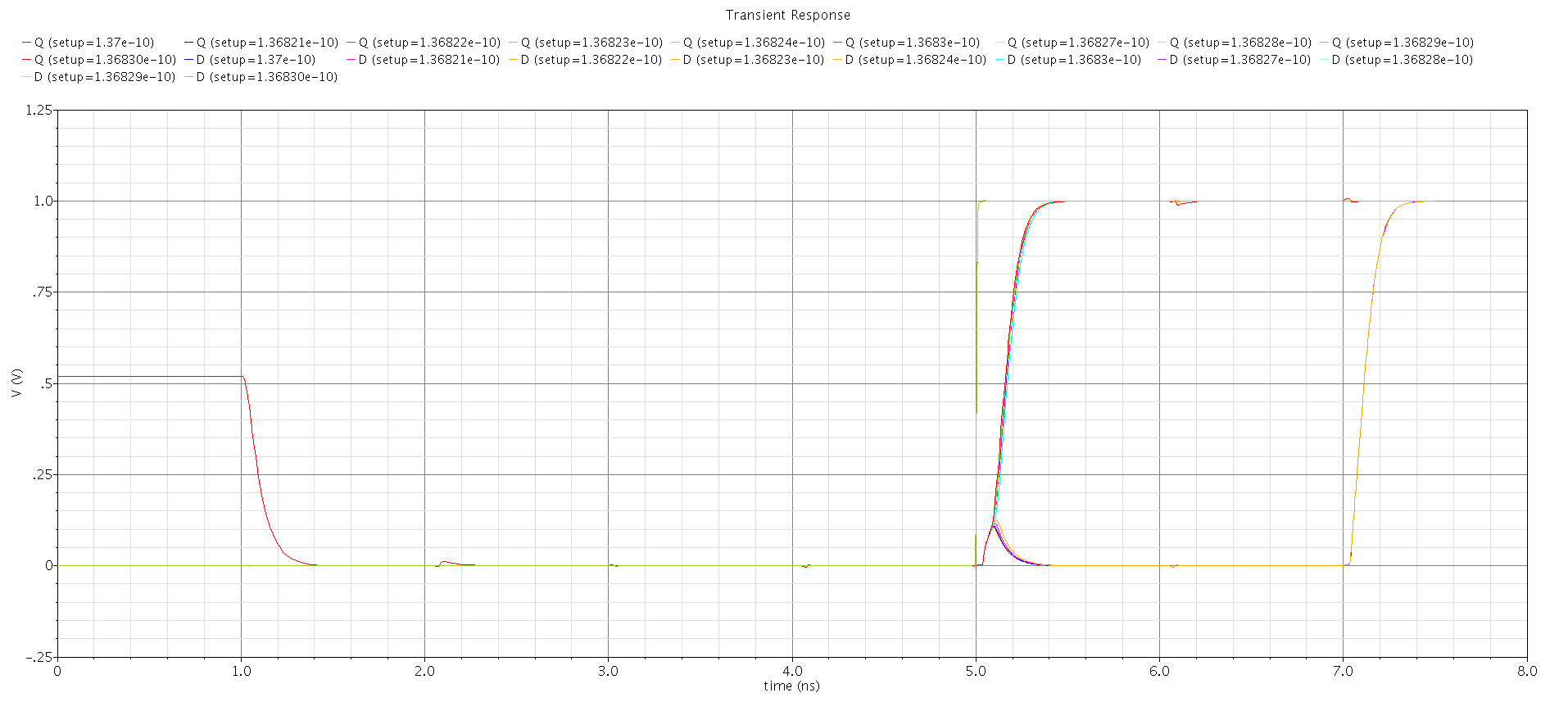
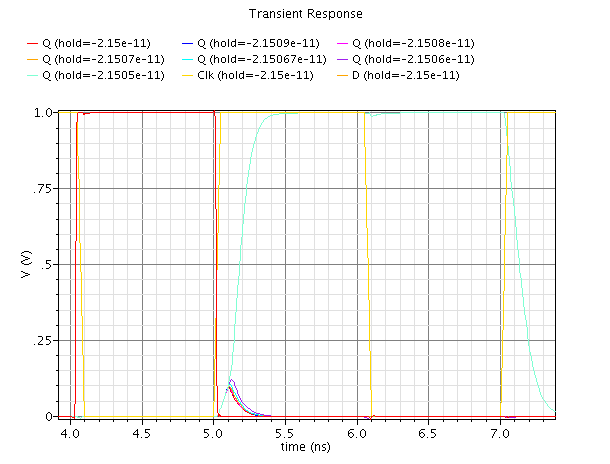
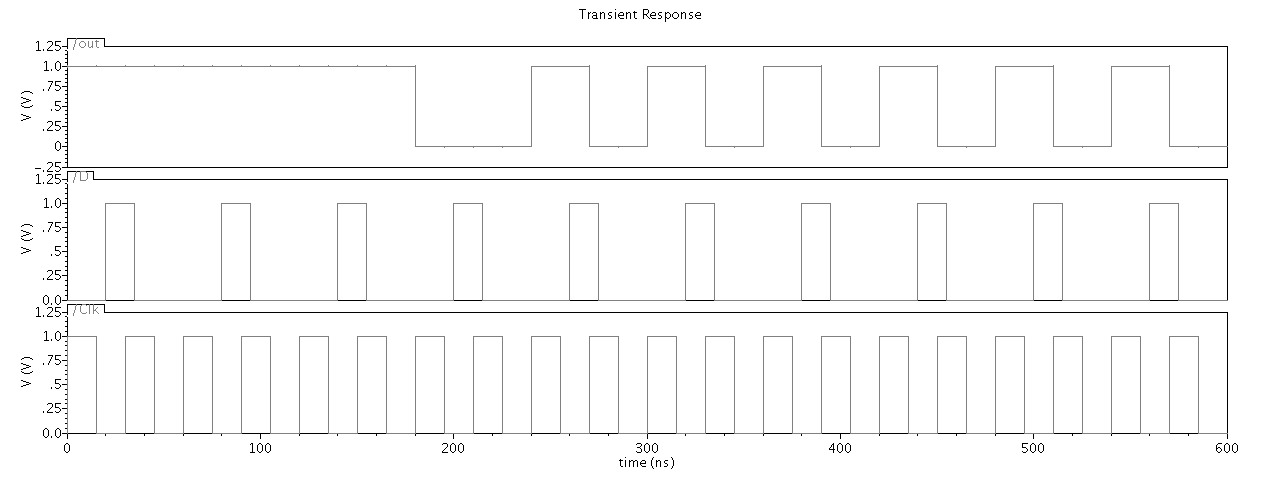
Figure : overlapping simulations of setup time

Figure 22 shows a group of parametric simulations approaching the setup time. As you can see some of the inputs approach Vm and return to 0V while others successfully transition to 1V. The presence of inputs approaching Vm means the parametric simulation was done with sufficient accuracy.

Figure : A simulation for calculating hold time

Calculating the hold time requires a simulation containing an input setting to 1V that does not violate setup time and transitioning it to 0V in a parametric simulation with a graph looking similar to that of the setup time. The successful transitions can be seen next to the Q values that approach Vm but return to 0V. This data is used to find the hold time, which we found to be -24 ps.

Figure : A simulation of the shift register

The simulation of the shift register begins with input of 0s and 1s into the register. After 8 clock cycles the first entered bit (1) is shown in Q. You can see the consecutive 0s and 1s shift out with continued clock cycles.